안전한 오라클 데이터베이스 운영을 위한 Check List

2002. 11. 22
오규철 연구원/해킹바이러스상담지원센터
kcoh@certcc.or.kr

<목 차>

1. 머릿말

가장 중요한 정보들은 어디에 있을까? 물론 시건장치가 된 금고속에 있는 정보도 중요하겠지만, 방대한 중요 정보들의 대부분은 데이터베이스에 기록될 것이다.

파괴되면 가장 곤란한 정보들은 무엇일까? 물론 홈페이지가 변조되는 것도 큰 문제이겠지만, (예를 들어)고객의 결제정보를 저장한 DB 테이블이 삭제되다면 아주 곤란한 치지가 되지 않을까?

현재 가장 보안이 부족한 항목은 무엇일까? 물리적보안? 네트워크보안? 혹시 시스템보안? 그렇다면, (심지어 디폴트 패스워드를 변경하지 않은 채로 유영되는 사례가 비일비재한) 데이터베이스는 안전한 것인가?

본 고는 국내에서 많이 사용되고 있는 오라클 데이터베이스에 대한 간단한 보안 체크리스트로, 특히 데이터베이스에 익숙하지 않은 보안담당자에게 일종의 가이드라인을 제공하기 위해 작성되었다.

본 고가 데이터베이스 보안에 다소간이라도 도움이 되었으면 한다.

※ 본 고의 내용은 UNIX 시스템을 기반으로 한 오라클을 기준으로 작성하였다.

※ 본 고에서 사용자가 직접 typing해야하는 명령어는 *색을 반전시킨 기울임체*로 작성하여 알아보기 쉽게 하였다.

2. 오라클 기본지식 - 본인을 만나기 전 미리 알아야 할 사항들

본 장에서는 아래와 같은 개념에 대해 설명하고자 한다.

- “데이터 딕셔너리(Data Dictionary)”의 개념
- 파라메터 파일
- 컨트롤 파일
- 오라클 네트워킹 인프라스트럭쳐 - Oracle Net Service
- 오라클 리스너
- privilege GRANT
- 오라클 데이터베이스에서 英文 대소문자의 구분
- "롤(role)"의 개념

안전한 데이터베이스의 운영을 위해서 알고 있어야 할 내용은 대단히 방대할 것이다. 그러나, 그 모든 것을 여기에서 언급할 수 없으므로, 본 고를 읽고 이해하는데 꼭 필요한 개념들만을 간략히 정리하여 보았다. 오라클 데이터베이스에 익숙한 독자라면 본 장을 생략하여도 무방하다.

가. 데이터 딕셔너리(Data Dictionary)

"데이터 딕셔너리(Data Dictionary)"는 오라클 데이터베이스의 가장 핵심적인 요소중에 하나로서, ‘데이터베이스 자신에 관한 정보’를 담고있으며, 구체적으로 아래와 같은 정보들을 제공한다.

- 데이터베이스의 물리적/논리적 구조
- 객체에 대한 정의 및 공간할당
- 사용자
- 롤(Roles)
- 권한(Privileges)
- 감사(Auditing)

데이터 딕셔너리는 테이블이다. 테이블인데, 소유자가 "sys"이며, "시스템 테이블 스페이스(System Table Space)"에 저장되어 있다. 또한, 오라클 서버에 의해서 자동으로 유지보수 된다. 즉, 사용자는 데이터 딕셔너리의 내용을 수

정할 수 없다.

일반적으로 데이터 딕셔너리는 "user", "all", "dba"와 같은 특정 문자열로 시작하는 테이블이므로 구분하기가 쉽다.

아래는 데이터베이스 사용자 현황을 알기위한 SQL문으로 데이터 딕셔너리인 "all_users"를 검색하고 있다.

```
SQL> SELECT * FROM all_users;

USERNAME                          USER_ID CREATED
------------------------------ ---------- ---------
SYS                                     0 15-SEP-00
SYSTEM                                  5 15-SEP-00
OUTLN                                  11 15-SEP-00
DBSNMP                                 16 15-SEP-00
TRACESVR                               19 15-SEP-00
AURORA$JIS$UTILITY$                    26 16-SEP-00
OSE$HTTP$ADMIN                         27 16-SEP-00
AURORA$ORB$UNAUTHENTICATED             28 16-SEP-00
ORDSYS                                 29 16-SEP-00
ORDPLUGINS                             30 16-SEP-00
MDSYS                                  31 16-SEP-00

USERNAME                          USER_ID CREATED
------------------------------ ---------- ---------
CTXSYS                                 34 16-SEP-00
SCOTT                                  36 16-SEP-00
ADAMS                                  37 16-SEP-00
JONES                                  38 16-SEP-00
CLARK                                  39 16-SEP-00
BLAKE                                  40 16-SEP-00
SECURE                                 45 08-AUG-01
BKJEON                                 46 04-DEC-01

19 rows selected.
```

나. 파라메터 파일(Parameter File)

"파라메터 파일(Parameter File)"은 오라클 데이터베이스 인스턴스의 초기화 설정에 사용되며, 오라클 데이터베이스 인스턴스가 시작될 때 읽혀진다.

파라메터 파일에는 다음과 같은 내용들을 포함하여, 데이터베이스의 전반적인 설정들이 기록되어 있다.

- 데이터베이스의 이름
- SGA(System Global Area)의 크기
- "컨트롤 파일(Control File)" 및 "어카이브 파일"의 이름과 위치
- 데이터베이스 블록의 크기

파라메터 파일의 경로 및 이름은 일반적으로

$ORACLE_HOME/dbs/init<sid>.ora

혹은

$ORACLE_HOME/dbs/spfile<sid>.ora

가 된다. 참고로 <sid>는 System ID를 의미하며, 오라클 데이터베이스 인스턴스(Instance)를 나타낸다.

파라메터 파일은 텍스트형식의 PFILE과 바이너리형식의 SPFILE 두 가지가 있는데, 전자는 OS가 제공하는 에디터로 수정하며, 데이터베이스를 재기동하여야 수정내용이 적용된다. 후자는 ALTER SQL 문을 사용하여 데이터베이스 인스턴스의 기동중에 변경할 수 있다.

앞에서 언급한 파라메터 파일 중 init<sid>.ora는 PFILE이고 spfile<sid>.ora는 SPFILE이다. 참고로 SFILE은 오라클9i에서 새롭게 등장한 개념으로, 오라클8i에는 적용되지 않는다.

init<sid>.ora와 spfile<sid>.ora 두가지가 모두 존재한다면spfile<sid>.ora의 내용이 우선순위를 갖는다.

설정된 파라메터를 확인하는 방법에는 아래와 같은 두가지가 있다.

SQL> *SHOW PARAMETER;*

NAME	TYPE	VALUE
O7_DICTIONARY_ACCESSIBILITY	boolean	TRUE
active_instance_count	integer	
always_anti_join	string	NESTED_LOOPS
always_semi_join	string	standard
aq_tm_processes	integer	0
audit_file_dest	string	?/rdbms/audit
audit_trail	string	NONE

.....
<출력내용 생략>
.....

SQL> *SELECT name FROM V$PARAMETER;*

NAME
---
processes
sessions
timed_statistics
timed_os_statistics
resource_limit
license_max_sessions
license_sessions_warning
cpu_count
instance_groups
event
shared_pool_size
shared_pool_reserved_size
.....
<출력내용 생략>
.....

다. 컨트롤 파일(Control File)

"컨트롤 파일(Control File)"은 데이터베이스의 물리적 상태를 정의하는 바이너리 형식의 파일로서, 다음과 같은 정보가 저장되어 있다.

- 데이터베이스 名(SID : System ID)
- 데이터 파일과 REDO 로그 파일 名
- 現 로그 순서 번호(Log Sequence number)
- 체크 포인트(Check Point) 정보

컨트롤 파일의 경로는 파라메터 파일내에 아래와 같은 형식으로 명시되어 있다.

```
control_files = ("/user1/oracle/OraHome1/oradata/ORA817/control01.ctl",
"/user1/oracle/OraHome1/oradata/ORA817/control02.ctl",
"/user1/oracle/OraHome1/oradata/ORA817/control03.ctl")
```

※ 단, "$ORACLE_HOME = /user1/oracle/OraHome1"인 경우임

컨트롤 파일이 손상되면, 데이터베이스의 기동이 불가능하다. 따라서, 위의 예에서처럼 여러 개의 컨트롤 파일(물론 각각의 내용은 동일하다.)을 지정하여 손상에 대비하는 것이 일반적이다.

컨트롤 파일의 내용을 보거나, 백업을 받으려면 아래의 명령어를 이용한다.

ALTER DATABASE BACKUP CONTROLFILE TO TRACE;

컨트롤 파일의 내용은 파라메터 파일내에 user_dump_dest로 지정된 디렉토리 안에 txt 파일로 저장된다. 앞에서 말한 것과 같이 컨트롤 파일은 바이너리 형식이므로, cat 등의 OS 명령으로 살펴 볼 수가 없다.

컨트롤 파일의 문제가 발생한 경우, 이렇게 txt로 백업받은 내용을 이용하여, 컨트롤 파일을 재생성 할 수 있다. 해당 txt파일의 내용을 살펴보면, 데이터베이스 파일이 No mount된 상태에서 컨트롤 파일을 생성하는 SQL문으로 구성되어 있음을 알 수 있다.

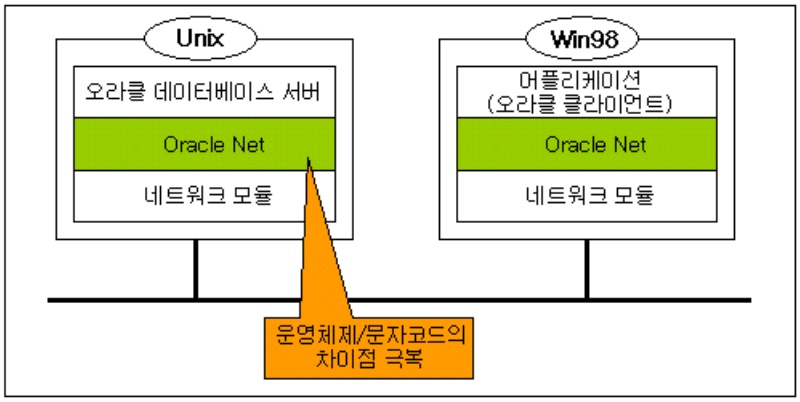
라. 오라클 네트워킹 인프라스트럭쳐 - Oracle Net Service

"Oracle Net Service"는 오라클 데이터베이스를 네트워크에서 사용할 때 기반이 되는 소프트웨어로, 어플리케이션과 오라클 데이터베이스의 중간에 위치하는 일종의 미들웨어이다.

"Oracle Net Service"는 오라클 데이터베이스와 오라클을 이용하는 클라이언트 사이에서 접속/단절, SQL query 및 query 결과의 통신 등을 수행한다.

예를 들어, Win98용 "Oracle Net Service"를 설치한 Win98 시스템의 어플리케이션은 UNIX 시스템 상에서 운영되는 오라클 데이터베이스에 (운영체제에 구애받지 않고)접근할 수 있다.

※ "Oracle Net Service"는 오라클9i부터 쓰이기 시작한 이름으로, 오라클8i 사용자들은 "Net8"이라는 용어에 더 친숙해 있다.



Oracle Net Service의 개념

마. 오라클 리스너

"리스너(Listener)"는 오라클 클라이언트로부터의 접속 요구를 감시하는 프로세스로서, 오라클 서버에서 작동한다.

클라이언트가 다른 호스트(데이터베이스 서버)에 존재하는 데이터베이스를 사용하고자 할 때, 해당 서버의 리스너를 통해 데이터베이스에 접속한다.

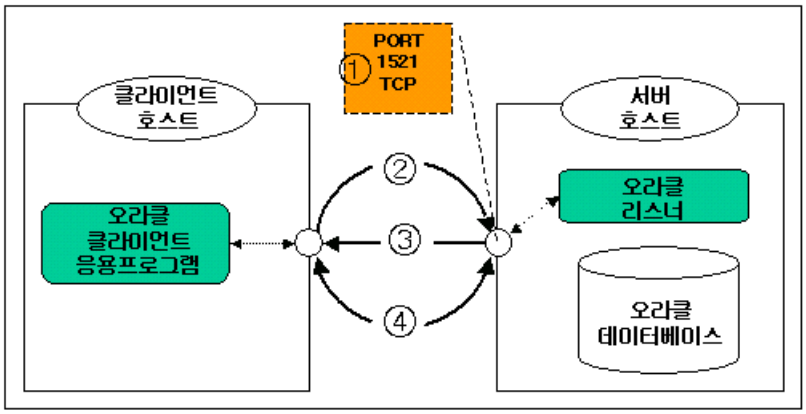
접속요구를 받은 리스너는 해당 접속요구가 올바른지를 판단하여, 올바른 경우 데이터베이스와 접속을 허가한다. 이상의 내용이 실행되는 순서는 아래와 같다.

① 서버의 리스너는 특정포트로 들어오는 접속요구를 감시한다.(디폴트 포트는 1521/TCP)

② 클라이언트는 접속하고자하는 데이터베이스의 리스너에게 접속요구를 송신한다. 원하는 데이터베이스는 네트워크 서비스명으로 지정한다.

③ 리스너는 수신한 접속요구가 올바른지를 분석한다. 해당 접속요구가 올바르면, 일련의 접속관련정보를 클라이언트에게 송신한다.

④ 클라이언트는 해당 접속관련정보를 이용하여 오라클 데이터베이스와 직접 통신한다.



오라클 리스너의 동작

리스너의 기동 및 정지에는 아래와 같이 lsnrctl 명령을 사용한다.

- 리스너의 기동 : *lsnrctl start*
- 리스너의 정지 : *lsnrctl stop*
- 리스너의 現상태 보기 : *lsnrctl status*

리스너는 "$ORACLE_HOME/network/admin/listener.ora" 라는 이름의 설정파일을 가지며, 해당 설정파일은 리스너의 기동시에 읽혀진다. 따라서 설정파일의 수정 후에는 리스너를 재기동 시켜야 한다.

바. privilege GRANT

"GRANT(권한부여)"란 생성된 사용자 아이디에 권한을 부여하는 것을 의미한다.

GRANT 명령으로 일반사용자에게 부여할 수 있는 권한은

- CONNECT 권한 : 데이터베이스에 접속 및 조회를 할 수 있는 권한
- RESOURCE 권한 : 데이터베이스의 자원을 사용할 수 있는 권한

이 있다. 동일한 방법으로 데이터베이스 관리자 계정에는 DBA권한을 GRANT한다.

이해를 돕기위해 사용자를 하나 생성하여 데이터베이스에 접속하는 아래와 같은 예를 보자.

```
줄번호 명령어
--------------------------------------------------------------------------------
(01)    SQL> CREATE USER kim IDENTIFIED BY kim_passwd;
(02)
(03)    User created.
(04)
(05)    SQL> CONNECT kim/kim_passwd;
(06)    ERROR:
(07)    ORA-01045: user KIM lacks CREATE SESSION privilege; logon denied
(08)
(09)
(10)    Warning: You are no longer connected to ORACLE.
(11)    SQL> CONNECT system/root_passwd
(12)    Connected.
(13)    SQL> GRANT CONNECT TO kim;
(14)
(15)    Grant succeeded.
(16)
(17)    SQL> CONNECT kim/kim_passwd;
(18)    Connected.
(19)    SQL>
```

※ 실제화면에서는 줄번호가 보이지않음.

줄번호 (01)에서 kim이라는 새로운 사용자 아이디를 생성하였다. 그러나, 줄번호 (05)~(07)를 보면 해당 아이디에 대한 데이터베이스 접속이 거부되었음을 알 수 있다. 줄번호 (13)에서 kim에 접속에 대한 '권한을 부여'(GRANT)한 후에야, 데이터베이스로의 접속이 가능(줄번호 (17),(18))하다.

사. 오라클 데이터베이스에서 英文 대소문자의 구분

오라클 데이터베이스는 대소문자를 구분하지 않는다. 그러나, 대소문자를 구분하는 것이 데이터베이스 성능에 영향을 줄 수 있다.

오라클 데이터베이스에는 Library Cache라는 개념이 있어서, 입력된 SQL 및 PL/SQL 문장의 parse 정보를 유지하여, 동일한 문장이 다시 들어올 때 재활용한다. 그런데, 이때 대소문자까지 완전히 동일한 문장만을 인식하므로, 대소문자의 구분이 성능에 영향을 줄 수 있다.

아. "롤(role)"의 개념

"롤(role)"이란 권한에 대한 일종의 그룹으로서, 롤을 활용하여 일괄적인 권한 부여(GRANT) 및 회수(REVOKE) 등을 행할 수 있다. 아래의 예를 보면 롤의 개념이 좀 더 명확해 질 것이다.

```
번호줄  명령
-------------------------------------------------------------------------------
(01)    SQL> CONNECT sys/pass_sys
(02)
(03)    SQL> CREATE ROLE newrole;
(04)
(05)    SQL> GRANT CONNECT TO newrole;
(06)
(07)    SQL> CONNECT scott/tiger
(08)
(09)    SQL> GRANT SELECT, DELETE ON emp TO newrole;
(10)
(11)    SQL> CONNECT sys/sys
(12)
(13)    SQL> CREATE USER abc IDENTIFIED BY pass_abc DEFAULT TABLESPACE user_data;
(14)
(15)    SQL> GRANT newrole TO abc;
(16)
(17)    SQL> CONNECT abc/pass_abc
(18)
(19)    SQL> SELECT * FROM scott.emp;
```

```
(20)
(21)    EMPNO ENAME      JOB              MGR HIREDATE          SAL       COMM      DEPTNO

(22)    --------- ---------- --------- --------- ----------------- --------- -----------
(23)    7369 SMITH       CLERK            7902 80/12/17          800                 20
(24)
(25)    7902 FORD        ANALYS           7566 81/12/03          3000                20
(26)    7934 MILLER      CLERK            7782 82/01/23          1300                10
(27)
(28)    14 개의 행이 선택되었습니다.
```

※ 실제화면에서는 줄번호가 보이지않음.

번호줄 (01)에서 DBA 권한으로 접속하였다. "롤"의 생성은 DBA로만 가능하다.

번호줄 (03)에서 newrole이란 이름의 새로운 "롤(Role)"을 생성하였다.

번호줄 (05)에서 newrole에게 CONNECT 권한 부여 (GRANT)하였다.

번호줄 (07)에서 사용자 scott로 접속하여, 번호줄 (09)에서emp 테이블에 대해 SELECT, DELETE 할 수 있는 권한을 부여하였다. 즉, 앞으로 newrole을 권한부여(GRANT)받는 사용자는 scott소유의 emp 테이블에 대해 SELECT, DELETE를 할 수 있다.

번호줄 (13)에서 DBA 권한으로 접속, abc란 사용자 아이디를 추가하였다.

번호줄 (15)에서 사용자 abc에게 newrole이란 이름의 롤을 권한 부여하였다.

이후 사용자 abc는 scott소유의 emp란 테이블에 SELECT, DELETE 가능하다.

번호줄 (21)~(28)은 scott소유의 emp란 테이블을 조회한 결과이다.

3. 오라클 보안 체크리스트

본 장에서는 오라클 데이터베이스의 설정오류로 인하여 발생할 수 있는 문제점을 체크리스트 형식으로 나열하였다.

이하의 내용은, 오라클사가 대중에게 제공하는 백서(White Paper)인 'A Security Checklist for Oracle 9i'를 근간으로 하고 있음을 밝혀둔다.

※ 해당 백서는 http://otn.oracle.co.kr/에서 구할 수 있다.

그럼, 다소 고루하기는 하나 가장 기본적인 내용을 필두로 본 장을 시작하겠다.

가. 꼭 필요한 것만 설치하여야 한다.

오라클 데이터베이스를 처음 설치할 때, 꼭 필요한 요소만 설치하여야 한다. 무엇이 꼭 필요한 요지인지 확실치 않다면, 일반적인 구성으로 설치(즉, Typical installation을 선택하여 설치)하라.

나. 디폴트 사용자 아이디들을 잠그고(lock) 기간만료(expire)시켜야 한다.

오라클 데이터베이스를 설치하면 다수의 디폴트 사용자 아이디가 생긴다. 이때 오라클의 사용자관리도구(DBCA : Database Client Administration Tool)가 이러한 디폴트 사용자 아이디를 자동으로 잠그고 기간만료 시키는데 (언제나 그렇듯이)예외가 되는 사용자 아이디들이 있다.

그 예외들은 아래와 같다.

SYS, SYSTEM, SCOTT, DBSNMP, OUTLN, 그리고 3개의 JSERV사용자 아이디들

만약 수동으로(즉, DBCA를 사용하지않고) 오라클을 설치하는 경우라면, 디폴트 사용자 아이디는 모두 열린(즉, lock되지 않는다.) 상태가 되므로 보다 세심한 주의가 필요하다. 따라서 수동으로 설치한 데이터베이스는 SQL문을 통하여 디폴트 사용자 아이디를 잠그고 기간만료 시켜야 한다. (물론 상기의 SYS, SYSTEM, SCOTT, DBSNMP, OUTLN, 그리고 3개의 JSERV사용자 아

이디들을 제외하고 말이다.)

다음과 같은 SQL을 통해서 사용자 아이디목록과 그 상태를 알 수 있다.

SQL> *SELECT username, account_status FROM dba_users;*

```
USERNAME                       ACCOUNT_STATUS
------------------------------ --------------------------------
SYS                            OPEN
SYSTEM                         OPEN
OUTLN                          OPEN
DBSNMP                         OPEN
TRACESVR                       OPEN
AURORA$JIS$UTILITY$            OPEN
OSE$HTTP$ADMIN                 OPEN
AURORA$ORB$UNAUTHENTICATED     OPEN
ORDSYS                         OPEN
ORDPLUGINS                     OPEN
MDSYS                          OPEN

USERNAME                       ACCOUNT_STATUS
------------------------------ --------------------------------
USER1                          OPEN
USER2                          OPEN
```

특정 사용자 아이디를 잠그고 기간만료 시키는 SQL 문장은 다음과 같다.

SQL> *ALTER USER test ACCOUNT LOCK PASSWORD EXPIRE;*

※ 단, External 인증의 사용하는 경우는 expire 할 수 없다.

또한 불필요한 사용자 아이디를 삭제하는 것도 좋은 방법이 될 것이다. 예를 들어 USER1을 삭제하고자 한다면 아래의 SQL 문장을 이용할 수 있다.

SQL> *DROP USER user1;*

다. '나.'의 단계에서 잠그고 기간만료하지 않은 디폴트 사용자 아이디(SYS, SYSTEM, SCOTT, DBSNMP, OUTLN, 그리고 3개의 JSERV 사용자 아이디)들의 암호를 변경시켜야 한다.

오라클 데이터베이스를 공격하는 가장 손쉬운 (또한 가장 어처구니없는)방법은 설치당시의 디폴트 암호를 사용하는 사용자 아이디를 찾아내는 것이다. 이러한 암호의 변경은 설치직후 지체없이 이루어져야 한다.

이 시점에서 현재 자신의 조직이(혹은 회사가) 운영하고 있는 오라클 데이터베이스의 보안상태가 궁금할 수 있을 것이다.

아래의 예제를 참조하여 데이터베이스에 접속해보자. SQLPLUS 라는 프로그램을 이용하면 데이터베이스로의 접속이 가능하다.

```
줄번호 명령
----------------------------------------------------------------------------------------------------
(01)    $ sqlplus
(02)
(03)    SQL*Plus: Release 8.1.7.0.0 - Production on Mon Sep 2 13:59:11 2002
(04)
(05)    (c) Copyright 2000 Oracle Corporation.  All rights reserved.
(06)
(07)    Enter user-name: system
(08)    Enter password: manager
(09)
(10)    Connected to:
(11)    Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
(12)    With the Partitioning option
(13)    JServer Release 8.1.7.0.0 - Production
(14)
(15)    SQL> exit
```

※ 실제화면에서는 줄번호는 보이지않음.

이때, SYSTEM(사용자 아이디) / MANAGER(암호)로 접속이 가능한가?

※ 오라클 데이타베이스는 대소문자 구분을 하지 않으므로, system/manager도 동일한 사용자 아이디이다.

접속이 가능하다면 해당 데이터베이스는 (디폴트 사용자 아이디를 알고만 있다면)누구라도 접근하여 파괴할 수 있다는 의미이다. 따라서 본 고를 더 읽기 전에 데이터베이스 관리자에게 무언가 한마디 충고라도 해주는 것이 좋을 것이다.

디폴트 사용자 아이디의 암호는 아래 표와 같다.

사용자 아이디종류	사용자 아이디	패스워드
관리자사용자 아이디	SYS	CHANGE_ON_INSTALL
	SYSTEM	MANAGER
일반사용자 아이디	SCOTT	TIGER
jserv 사용자 아이디	AURORAJISUTILITY$	임의로 생성된 패스워드
	OSE$HTTP$ADMIN	임의로 생성된 패스워드
	AURORAORBUNAUTHENTICATED	임의로 생성된 패스워드

※ 그 외의 디폴트 사용자 아이디는 사용자 아이디와 암호가 동일함 ex) MDSYS / MDSYS

아래의 예제는 user2 라는 사용자 아이디의 암호를 'new_passwd'로 변경하는 SQL 문장이다. 다른 디폴트 사용자 아이디도 동일한 방법으로 변경할 수 있다.

SQL> *ALTER USER user2 IDENTIFIED BY new_passwd;*

또한, 오라클 엔터프라이즈 에디션을 사용한다면 kerberos, 토큰 카드(token card), 스마트 카드, X.509 인증서 등과 같은 강화된 인증기능을 이용할 수 있다.

라. "데이터 딕셔너리(Data Dictionary)"를 보호해야 한다.

"데이터 딕셔너리(Data Dictionary)"를 보호하기 위해서는 "파라메터 파일(Parameter File)"인 init<sid>.ora의 내용을 OS가 제공하는 에디터를 이용하여 아래와 같이 수정하면 된다.

```
O7_DICTIONARY_ACCESSIBILITY = FALSE
```

이렇게하면 오직 적절한 권한을 가진 사용자(즉, DBA 권한으로 접속을 생성

한 사용자)만이 "데이터 딕셔너리" 상의 'ANY' 시스템권한('ANY' system privilege)를 사용할 수 있다.

※ 참고로 DBA 권한으로 접속을 맺으려면 'CONNECT /AS SYSDBA'라는 명령어를 사용하면 된다.

만일 이러한 설정을 위처럼 하지 않는다면, 'DROP ANY TABLE' 시스템 권한을 가진 사용자는 누구라도 "데이터 딕셔너리"의 내용을 악의적으로 DROP할 수 있을 것이다.

"데이터 딕셔너리"를 조회해야만 하는 사용자에게는 'SELECT ANY DICTIONARY' 시스템 권한을 주어 "데이터 딕셔너리" 뷰(view)로의 접근만을 허용하는 유두리있는 방식을 이용할 수 있다.

오라클9i에서는 디폴트로 O7_DICTIONARY_ACCESSIBILITY = FALSE 값을 갖는다. 그러나 오라클8i에서는 해당 값이 디폴트로 TRUE로 설정되어 있으므로 반드시 수정하여야 한다.

마. 권한(privilege)의 부여(GRANT)와 관련된 사항 - 꼭 필요한 만큼만 권한을 주어야 한다.

① 사용자들에게 꼭 필요한 최소권한(least privilege)만을 부여(GRANT)하여야 한다.

② PUBLIC 사용자 그룹에서 불필요한 권한을 회수(REVOKE)하여야 한다.

PUBLIC은 오라클 데이터베이스의 모든 사용자에게 디폴트 롤(role)로 적용된다. 따라서 모든 사용자는 PUBLIC에 권한 부여(GRANT)된 것은 어떤 일이든 할 수 있다. 이런 경우 사용자가 교묘하게 선택된 PL/SQL 패키지를 실행시켜 본래 자신에게 권한 부여된 권한 범위를 넘어서는 작업을 할 수도 있을 것이다.

③ 또한 PL/SQL 보다 더 강력한, 아래와 같은 패키지들도 오용될 소지가 있으므로 주의하여야 한다.

패키지명	패키지의 역할	발생할 수 있는 문제점
UTL_SMTP	임의의 메일 메시지를 임의의 사용자간에 전송할 수 있도록 하는 패키지.	이 패키지를 PUBLIC 그룹에서 사용할 수 있도록 권한부여(GRANT)하면 허가받지 않은 메일전송이 발생할 수 있음.
UTL_TCP	외부의 네트워크 서비스로 TCP 컨넥션을 열 수 있도록 하는 패키지.	임의의 데이터가 데이터베이스 서버와 외부의 네트워크 서비스 사이에서 오갈 수 있음.
UTL_HTTP	HTTP를 통한 데이터 검색 등을 가능케하는 패키지.	HTML 형식의 임의의 데이터가 전송될 수 있음
UTL_FILE	파일처리와 관련된 패키지	설정이 잘못되는 경우, 정보시스템상의 모든 파일에 TXT LEVEL의 접근이 가능할 수 있음.
DBMS_RANDOM	저장된 데이터를 암호화하는데 사용되는 패키지	일반적으로 대부분의 사용자들은 데이터를 암호화하는 권한을 가져서는 안됨.

이와같은 패키지들은 특정한 응용프로그램에 아주 유용하게 이용될 수 있다. 바꾸어 말하면, 모든 경우에 이러한 패키지들을 꼭 필요로 하는 것이 아니라는 듯이다. 꼭 필요하지 않은 패키지들의 사용권한을 PUBLIC에서 제거하자.

④ 'run-time facilities'에 제약된 퍼미션을 주어야 한다. (Restrict permission on run-time facilities).

'오라클 자바 버추얼 머신(OJVM : Oracle Java Virtual Machine)'이 데이터베이스 서버의 run-time facility의 예가 될 수 있다. 어떠한 경우라도 이러한 run-time facility에 'all permission'을 주어서는 안된다.

또한 데이터베이스 서버 외부에서 파일이나 패키지를 실행할 수 있는 facility에 어떤 퍼미션을 줄 때는 반드시 정확한 경로를 명시하여야 한다. 아래의 예를 자세히 살펴보면 좀 더 이해가 쉬울 것이다.

- 취약한 run-time call의 예제

```
call dbms_java.grant_permission('SCOTT','SYS:java.io.FilePermission','<<ALL FILES>>','read');
```

- 안전한 run-time call의 예제

```
call dbms_java.grant_permission('SCOTT','SYS:java.io.FilePermission','<<actual directory path >>','read');
```

바. 강력한 인증정책을 수립하여 운영하여야 한다.

① 클라이언트에 대한 철저한 인증이 필요하다.

오라클 9i는 원격인증 기능을 제공한다. 만일 해당기능이 활성화되면(TRUE), 원격의 클라이언트들이 오라클 데이터베이스에 접속할 수 있도록 한다. 즉, 데이터베이스는 적절하게 인증된(즉, 클라이언트 자체의 OS가 인증한) 모든 클라이언트들을 신뢰한다. 주의하라. 일반적으로 PC의 경우에는 적절한 인증여부를 보장할 수 없다. 따라서 원격인증 기능을 사용하면 보안이 대단히 취약해진다.

원격인증기능을 비활성화(FALSE)하도록 설정한다면 오라클 데이터베이스에 접속하려는 클라이언트들은 server-based 인증(즉, 데이터베이스 서버의 의증)을 해야하므로 보안이 강화된다.

원격인증을 제한하여 클라이언트의 인증을 데이터베이스 서버가 행하도록 하려면 오라클 "파라메터 파일(Parameter File)"인 init<sid>.ora의 내용을 OS가 제공하는 에디터를 이용하여 아래와 같이 수정하면 된다.

```
REMOTE_OS_AUTHENTICATION = FALSE
```

② 데이터베이스 서버가 있는 시스템의 사용자 수를 제한하여야 한다.

오라클 데이터베이스가 운영되고 있는 시스템의 사용자수를 OS 차원에서 제한하여야 한다. 제한이란 꼭 필요한 사용자 아이디만 만들라는 의미로서 관리자권한을 가진 사용자에 특히 주의해야 함은 두말할 것도 없다.

※ 오라클사(Oracle Corporation)는 백서(White Paper)인 'A Security Checklist for Oracle 9i'에서 시스템 관리자, 해당 데이터베이스의 소유자 혹은 그 누구라도 오라클 데이터베이스의 홈디렉토리 아래의 디폴트화일이나 디렉토리 퍼미션을 오라클사의 지도없이 변경하지 말 것을 권고하고 있다.

사. 네트워크를 통한 접근을 제한하라.

① 방화벽을 구축/운영하라.

다른 중요한 서비스와 마찬가지로 데이터베이스 서버는 방화벽 뒤에 설치하여야 한다. 오라클 네트워킹 인프라스트러쳐인 Oracle Net Service (Net8 and SQL*Net으로 많이 알려져 있다.)는 다양한 종류의 방화벽을 지원한다.

② 어렵게 방화벽을 구축하였다면 허점을 만들지 말라.

해커가 아니라면 누가 일부러 방화벽에 허점을 만들까? 그러나, 오라클 데이터베이스를 외부 네트워크에서 접근할 수 있도록 방화벽의 1521 port를 open 하며 스스로 치명적인 허점을 만드는 경우가 있을지도 모른다.

더 나아가, 암호설정 없이 오라클 리스너를 운영한다면 데이터베이스에 대한 중요한 정보(trace & loggin 정보, banner information, db descriptor, service name)들이 노출될 수 있다. 이러한 노출정보가 많으면 많을수록 데이터베이스가 공격당할 가능성이 높아질 것이다.

③ 원격에서 오라클 리스너의 설정을 함부로 변경할 수 없도록 하여야 한다.

아래와 같은 형식으로 listener.ora(오라클 리스너 설정화일 : Oracel listener control file)내의 파라메터를 설정하면, 원격에서 오라클 리스너 설정을 함부로 바꿀 수 없게 된다.

```
ADMIN_RESTRICTIONS_listener_name=ON
```

④ 접속을 허용할 네트워크 IP 주소 대역을 지정하는 것이 좋다.

데이터베이스 서버가 특정한 IP 주소대역으로부터의 클라이언트 접속을 제어하려면 “Oracle Net valid node checking“ 기능을 이용하면 된다. 이 기능을 사용하려면 protocol.ora(Oracle Net configuration file)내의 파라메터를 아래와 같이 설정하여야 한다.

```
tcp.validnode_checking = YES
tcp.excluded_nodes = { list of IP addresses }
tcp.invited_nodes = { list of IP addresses }
```

직관적으로 알 수 있듯이 첫 번째 파라메터가 나머지 두 개 파라메터 기능의 활성화를 결정하며, invited_nodes에 포함된 IP 주소 대역의 접속 요구만이 받아들여진다.

※ 잘 생각해보면 이 기능은 DoS 공격의 잠재적인 위협도 경감시켜줌을 알 수 있다.

⑤ 네트워크 트래픽을 암호화하라.

가능하다면 'Oracle Advanced Security'를 사용하여, 네트워크 트래픽을 암호화하라. (문제는 Oracle Advanced Security가 오라클 데이터베이스 엔터프라이즈 에디션에서만 제공된다는 점이다.)

⑥ 데이터베이스 서버가 있는 시스템의 OS를 강화하라.

불필요한 서비스를 제거하면, 데이터베이스 서버 시스템이 보다 안전해진다. UNIX와 Windows를 막론하고 불필요한(그리고 보안취약점이 있는) 많은 서비스들을 디폴트로 제공한다.(ftp, tftp, telnet 등등)

또한 제거된 서비스가 사용하는 UDP/TCP port를 막아라. 이때, UDP/TCP port 둘중 하나만 막는 실수를 저지르기 쉽다.

아. 모든 보안 패치를 바로바로 적용하여야 한다.

오라클 데이터베이스가 운영되고 있는 OS와 데이터베이스 자신에 대한 모든 중요한 패치를 정기적으로 실시하여야 한다. 조직이나 기업 차원에서 패치와 관련된 업무 프로세스를 만드는 것도 좋다.

또한, 아래의 사이트에서 보안과 관련된 정보를 얻을 수 있을 것이다.

- http://otn.oracle.com
- http://technet.oracle.com

※ 오라클사는 자사 제품의 보안취약점을 발견하는 경우 SECALERT@ORACLE.COM로 연락해 줄 것을 부탁하고 있다.

4. 오라클관련 취약점 현황

아래는 2002년에 새로 발견된 오라클 관련 취약점 중 CERTCC-KR이 보안권고문을 작성·배포한 항목들을 요약한 것이다. 본 장만으로 오라클 데이터베이스와 관련된 모든 보안 문제를 살펴보았다고 하기는 어렵지만, 최근 어떠한 문제점들이 부각되고 있는지, 그 해결책들은 어떻게 주어지는지 등의 현황은 간단하게 살펴볼 수 있을 것이다.

※ 보안권고문과 관련된 상세한 내용은 http://www.certcc.or.kr를 참조

가. Oracle 8i/9i Listener SERVICE_CURLOAD 명령어의 DoS 공격 취약점 (KA-2002-86)

① 취약한 제품 버전

- Oracle 9i Release 2 (9.2.x)
- Oracle 9i Release 1 (9.0.x)
- Oracle 8i (8.1.x)

② 시스템에 미치는 영향

SERVICE_CURLOAD 명령어를 실행할 때 Oracle TNS Listener는 서비스 거부공격을 당할 수 있다.

③ 취약점 개요

오라클 리스너와 연결을 맺고 "(CONNECT_DATA=(COMMAND=SERVICE_CURLOAD))" 명령어를 실행하면 성공적 실행이라는 메시지와 함께 실행된다. 그러나, 연결을 종료를 하게되면 Listener도 서비스를 중단하게 된다. DoS공격은 이때 발생할 수 있으며, 또한 DoS 공격의 영향은 공격자가 얼마나 오랫동안 Listener와 연결을 지속하느냐에 달려 있다.

④ 해결책

- 패치적용
  http://metalink.oracle.com 참조
- 패치관련 정보는 아래의 문서를 참조
  http://otn.oracle.com/deploy/security/pdf/2002alert42rev1.pdf 참조

나. 오라클9i 사용자 권한 취약점 (KA-2002-040)

① 취약한 제품 버전

- Oracle9i Database, Release 9.0.1.x를 이용하는 모든 플랫폼

② 시스템에 미치는 영향

일반 사용자가 비인가된 데이터에 대한 접근을 할 수 있다.

③ 취약점 개요

이 취약점은, 일반사용자가 outer join의 SQL문을 이용하여 권한이 주어진 데이터에 접근할 수 있음에서 비롯된다. 이러한 취약점을 아는 사용자는 비인가된 접근을 통하여 오라클 데이터베이스 서버에 있는 자료를 유출할 수 있다.

④ 해결책

- 패치적용
  http://metalink.oracle.com.

※ "Patches" 버튼을 클릭하고 버그 치료 번호를 제출하면 된다. (버그번호는 2121935)

다. Oracle PL/SQL Apache 모듈의 다중 버퍼오버플로우 (KA-2002-030)

① 취약한 제품 버전

- Oracle 9iAS

② 시스템에 미치는 영향

임의의 명령어 실행 혹은 DoS 공격의 위험이 있다.

③ 취약점 개요

Oracle Application Server의 PL/SQL모듈에 다중 버퍼오버플로가 있다. 이로 인해 임의의 명령어를 실행시킬 수 있으며, 시스템이 DoS 공격을 받을 위험도 존재한다.

④ 해결책

- 패치적용
  http://metalink.oracle.com

라. Oracle 9iAS 환경설정 취약점 (KA-2002-025)

① 취약한 제품 버전

- Oracle 9iAS

② 시스템에 미치는 영향

PL/SQL모듈의 버퍼오버플로로 인하여 임의의 명령어 실행할 수 있으며, DoS 공격의 위험이 있다.

③ 취약점 개요

Oracle 9iAS를 사용한 웹 서비스는 Apache 서버에 의해서 구동되며 PL/SQL, SOAP(Simple Object Access Protocol), XSQL, JSP와 같은 웹 서비스를 제공할 수 있는 환경을 제공한다.

그런데, OracleJSP 환경에 보안 문제점이 있어서 공격자로 하여금 JSP 페이지의 소스코드에 대한 정보를 얻을 수 있게 한다. 또한 이런 문제점으로 인하여 공격자가 globals.jsa의 내용을 획득할 수 있다.

④ 해결책

문제점을 해결하기 위해서, $ORACLE_HOME$/apache/apache/conf 디렉토리 안에 있는 httpd.conf화일을 다음과 같이 편집해야 한다.

globals.jsa 화일에 대한 접근을 방지하기 위해선 다음의 entry를 추가한다.

```
Order allow,deny
Deny from all
```

.java 화일에 대한 접근을 방지하기 위해선 다음의 entry를 추가한다.

```
Order deny,allow
Deny from all
```

만약, JSP 페이지가 htdocs의 서브디렉토리에 저장되어 있지 않다면 다음

의 entry도 추가시켜야한다.

```
Order deny,allow
Deny from all
```

마. Oracle 9i Database Server 원격 취약점 (KA-2002-016)

① 취약한 제품 버전

- Oracle 9i, 8i 이 탑재되어 있는 모든 Operating System

② 시스템에 미치는 영향

오라클 데이터베이스 서버 라이브러리 내에 탑재되어 있는 함수를 임의로 인증절차 없이 실행시킬 수 있다.

③ 취약점 개요

오라클 데이터베이스의 많은 부분들이 PL/SQL 패키지에 의해 제공되어진다. PL/SQL는 SQL을 확장하여 Dynamic Link Library 또는 O/S 라이브러리의 함수를 호출할 수 있는 패키지로 구현되어져 있다.

공격자는 system() 함수를 호출하여 프로그램을 실행하고자 하면, 반드시 오라클 데이터베이스에 로그인 접속한 후 공격이 성공하기 이전에 CREATE LIBRARY에 대한 허가를 받아야 하는데 공격자는 인증절차 없이 임의의 라이브러리를 로딩할 수 있는 취약점이 발견되었다.

④ 해결책

첫째: Firewall 설정을 통해 리스너 포트로의 접근을 차단하라.

둘째: 오라클 데이터베이스의 PLSExtproc 기능이 필요 없다면 제거하라.

5. 결언 - 가상 시나리오

4장에서 살펴본 보안권고문들은 기술적인 취약점로 인해 시스템에 불법접근·자료유출·DoS 공격할 수 있는 문제를 다루고 있다. 해당 보안권고문의 내용을 참조하여, 데이터베이스의 보안을 강화하는 것은 대단히 중요하다. 그러나 이보다 더 중요한 사항 한가지를 언급하며 본고를 마무리하고자 한다.

다음과 같은 시나리오를 가정해보자.

A社는 전자상거래로 주요 수입을 내는 회사로 중요한 데이터베이스 정보에 대해 모든 transaction 로그를 남기는 방식으로 백업을 하고 있다. 해당 데이터베이스 관리자를 '갑'이라고 하자.

그런데, 어떤 이유로 '갑'은 갑자기 해고당하게 된다. '갑'은 분노를 참지못하고, 회사에 앙심을 품게된다.

고심하던 '갑'은 백업시스템과 동일한 이름의 dummy 파일을 만들어 /dev/null로 링크한다.(데이터베이스가 생성하는 모든 로그는 허공으로 사라지게 된다.)

또한 '갑'은 회사에 최대한 큰 타격을 주기위해, 6개월 후, 데이터베이스의 테이블이 모두 삭제하고, 관련 로그를 지우도록 한 부비트랩 스크립트를 작성하여 설치한다.

회사의 주요 업무를 관장하는 데이터베이스가 파괴된 후, 복구할 수 없다면(특히 거래내역 등, 금전과 관련된 정보를 망실하였다면)치명적인 타격을 입게될 것이다.

짐작대로 이 시나리오는 내부 보안의 중요성을 설파하기 위해 만들어졌다.

일반적으로 데이터베이스는 외부에서 접근할 수 없도록 설정되어 있으며, 또한, 외부인은 '어느것이 중요한 데이터인지' 구분하기 어려울 것이므로, 데이터베이스에 대한 정말 심각한 공격은 내부자에 의해 이루어질 가능성이 높다.

결국 이 모든 것은 기술적인 능력만으로 해결할 수 없고, 조직/기업이 조직적인 보안체계를 확립하여 데이터베이스 보안과 관련된 정책을

- 일관성있게
- 꾸준히

적용하여야 하는 것이다.

[참고자료]

1. A Security Checklist for Oracle9i(An Oracle white paper)

2. A3 Focus(A3 Security Consulting)

3. 양수정, 오라클 SQL Handbook, 2001. 4

4. Yamada Seiichi, Sugahara Tsuyoshi, Oracle DataBase, 2002. 5

5. 한국정보보호진흥원, 보안권고문, http://www.certcc.or.kr